Nivio Ziviani
Ricardo Baeza-Yates (Eds.)

String Processing and Information Retrieval

LNCS 4726

14th International Symposium, SPIRE 2007
Santiago, Chile, October 2007
Proceedings

Springer

Nivio Ziviani Ricardo Baeza-Yates (Eds.)

String Processing and Information Retrieval

14th International Symposium, SPIRE 2007
Santiago, Chile, October 29-31, 2007
Proceedings

Springer

Volume Editors

Nivio Ziviani
Federal University of Minas Gerais
Department of Computer Science
Av. Antônio Carlos 6627, 31270-010 Belo Horizonte, MG, Brazil
E-mail: nivio@dcc.ufmg.br

Ricardo Baeza-Yates
Yahoo! Research Latin America
Blanco Encalada 2120, Santiago 6511224, Chile
E-mail: ricardo@baeza.cl

Library of Congress Control Number: 2007937296

CR Subject Classification (1998): H.3, H.2.8, I.2, E.1, E.5, F.2.2

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN	0302-9743
ISBN-10	3-540-75529-2 Springer Berlin Heidelberg New York
ISBN-13	978-3-540-75529-6 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12171385 06/3180 5 4 3 2 1 0

Lecture Notes in Computer Science 4726

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Lecture Notes in Computer Science

Sublibrary 1: Theoretical Computer Science and General Issues

For information about Vols. 1– 4490
please contact your bookseller or Springer

Vol. 4782: R. Perrott, B.M. Chapman, J. Subhlok, R.F. de Mello, L.T. Yang (Eds.), High Performance Computing and Communications. XIX, 823 pages. 2007.

Vol. 4771: T. Bartz-Beielstein, M.J. Blesa Aguilera, C. Blum, B. Naujoks, A. Roli, G. Rudolph, M. Sampels (Eds.), Hybrid Metaheuristics. X, 202 pages. 2007.

Vol. 4770: V.G. Ganzha, E.W. Mayr, E.V. Vorozhtsov (Eds.), Computer Algebra in Scientific Computing. XIII, 460 pages. 2007.

Vol. 4763: J.-F. Raskin, P.S. Thiagarajan (Eds.), Formal Modeling and Analysis of Timed Systems. X, 369 pages. 2007.

Vol. 4746: A. Bondavalli, F. Brasileiro, S. Rajsbaum (Eds.), Dependable Computing. XV, 239 pages. 2007.

Vol. 4743: P. Thulasiraman, X. He, T.L. Xu, M.K. Denko, R.K. Thulasiram, L.T. Yang (Eds.), Frontiers of High Performance Computing and Networking ISPA 2007 Workshops. XXIX, 536 pages. 2007.

Vol. 4742: I. Stojmenovic, R.K. Thulasiram, L.T. Yang, W. Jia, M. Guo, R.F. de Mello (Eds.), Parallel and Distributed Processing and Applications. XX, 995 pages. 2007.

Vol. 4736: S. Winter, M. Duckham, L. Kulik, B. Kuipers (Eds.), Spatial Information Theory. XV, 455 pages. 2007.

Vol. 4732: K. Schneider, J. Brandt (Eds.), Theorem Proving in Higher Order Logics. IX, 401 pages. 2007.

Vol. 4731: A. Pelc (Ed.), Distributed Computing. XVI, 510 pages. 2007.

Vol. 4726: N. Ziviani, R. Baeza-Yates (Eds.), String Processing and Information Retrieval. XII, 311 pages. 2007.

Vol. 4711: C.B. Jones, Z. Liu, J. Woodcock (Eds.), Theoretical Aspects of Computing – ICTAC 2007. XI, 483 pages. 2007.

Vol. 4710: C.W. George, Z. Liu, J. Woodcock (Eds.), Domain Modeling and the Duration Calculus. XI, 237 pages. 2007.

Vol. 4708: L. Kučera, A. Kučera (Eds.), Mathematical Foundations of Computer Science 2007. XVIII, 764 pages. 2007.

Vol. 4707: O. Gervasi, M.L. Gavrilova (Eds.), Computational Science and Its Applications – ICCSA 2007, Part III. XXIV, 1205 pages. 2007.

Vol. 4706: O. Gervasi, M.L. Gavrilova (Eds.), Computational Science and Its Applications – ICCSA 2007, Part II. XXIII, 1129 pages. 2007.

Vol. 4705: O. Gervasi, M.L. Gavrilova (Eds.), Computational Science and Its Applications – ICCSA 2007, Part I. XLIV, 1169 pages. 2007.

Vol. 4703: L. Caires, V.T. Vasconcelos (Eds.), CONCUR 2007 – Concurrency Theory. XIII, 507 pages. 2007.

Vol. 4700: C.B. Jones, Z. Liu, J. Woodcock (Eds.), Formal Methods and Hybrid Real-Time Systems. XVI, 539 pages. 2007.

Vol. 4699: B. Kågström, E. Elmroth, J. Dongarra, J. Waśniewski (Eds.), Applied Parallel Computing. XXIX, 1192 pages. 2007.

Vol. 4698: L. Arge, M. Hoffmann, E. Welzl (Eds.), Algorithms – ESA 2007. XV, 769 pages. 2007.

Vol. 4697: L. Choi, Y. Paek, S. Cho (Eds.), Advances in Computer Systems Architecture. XIII, 400 pages. 2007.

Vol. 4688: K. Li, M. Fei, G.W. Irwin, S. Ma (Eds.), Bio-Inspired Computational Intelligence and Applications. XIX, 805 pages. 2007.

Vol. 4684: L. Kang, Y. Liu, S. Zeng (Eds.), Evolvable Systems: From Biology to Hardware. XIV, 446 pages. 2007.

Vol. 4683: L. Kang, Y. Liu, S. Zeng (Eds.), Advances in Computation and Intelligence. XVII, 663 pages. 2007.

Vol. 4681: D.-S. Huang, L. Heutte, M. Loog (Eds.), Advanced Intelligent Computing Theories and Applications. XXVI, 1379 pages. 2007.

Vol. 4672: K. Li, C.R. Jesshope, H. Jin, J.-L. Gaudiot (Eds.), Network and Parallel Computing. XVIII, 558 pages. 2007.

Vol. 4671: V.E. Malyshkin (Ed.), Parallel Computing Technologies. XIV, 635 pages. 2007.

Vol. 4669: J.M. de Sá, L.A. Alexandre, W. Duch, D. Mandic (Eds.), Artificial Neural Networks – ICANN 2007, Part II. XXXI, 990 pages. 2007.

Vol. 4668: J.M. de Sá, L.A. Alexandre, W. Duch, D. Mandic (Eds.), Artificial Neural Networks – ICANN 2007, Part I. XXXI, 978 pages. 2007.

Vol. 4666: M.E. Davies, C.J. James, S.A. Abdallah, M.D Plumbley (Eds.), Independent Component Analysis and Blind Signal Separation. XIX, 847 pages. 2007.

Vol. 4665: J. Hromkovič, R. Královič, M. Nunkesser, P. Widmayer (Eds.), Stochastic Algorithms: Foundations and Applications. X, 167 pages. 2007.

Vol. 4664: J. Durand-Lose, M. Margenstern (Eds.), Machines, Computations, and Universality. X, 325 pages. 2007.

Vol. 4649: V. Diekert, M.V. Volkov, A. Voronkov (Eds.), Computer Science – Theory and Applications. XIII, 420 pages. 2007.

Vol. 4647: R. Martin, M.A. Sabin, J.R. Winkler (Eds.), Mathematics of Surfaces XII. IX, 509 pages. 2007.

Vol. 4646: J. Duparc, T.A. Henzinger (Eds.), Computer Science Logic. XIV, 600 pages. 2007.

Vol. 4644: N. Azémard, L. Svensson (Eds.), Integrated Circuit and System Design. XIV, 583 pages. 2007.

Vol. 4641: A.-M. Kermarrec, L. Bougé, T. Priol (Eds.), Euro-Par 2007 Parallel Processing. XXVII, 974 pages. 2007.

Vol. 4639: E. Csuhaj-Varjú, Z. Ésik (Eds.), Fundamentals of Computation Theory. XIV, 508 pages. 2007.

Vol. 4638: T. Stützle, M. Birattari, H. H. Hoos (Eds.), Engineering Stochastic Local Search Algorithms. X, 223 pages. 2007.

Vol. 4630: H.J. van den Herik, P. Ciancarini, J. Donkers (Eds.), Computers and Games. XII, 283 pages. 2007.

Vol. 4628: L.N. de Castro, F.J. Von Zuben, H. Knidel (Eds.), Artificial Immune Systems. XII, 438 pages. 2007.

Vol. 4627: M. Charikar, K. Jansen, O. Reingold, J.D.P. Rolim (Eds.), Approximation, Randomization, and Combinatorial Optimization. XII, 626 pages. 2007.

Vol. 4624: T. Mossakowski, U. Montanari, M. Haveraaen (Eds.), Algebra and Coalgebra in Computer Science. XI, 463 pages. 2007.

Vol. 4623: M. Collard (Ed.), Ontologies-Based Databases and Information Systems. X, 153 pages. 2007.

Vol. 4621: D. Wagner, R. Wattenhofer (Eds.), Algorithms for Sensor and Ad Hoc Networks. XIII, 415 pages. 2007.

Vol. 4619: F. Dehne, J.-R. Sack, N. Zeh (Eds.), Algorithms and Data Structures. XVI, 662 pages. 2007.

Vol. 4618: S.G. Akl, C.S. Calude, M.J. Dinneen, G. Rozenberg, H.T. Wareham (Eds.), Unconventional Computation. X, 243 pages. 2007.

Vol. 4616: A.W.M. Dress, Y. Xu, B. Zhu (Eds.), Combinatorial Optimization and Applications. XI, 390 pages. 2007.

Vol. 4614: B. Chen, M. Paterson, G. Zhang (Eds.), Combinatorics, Algorithms, Probabilistic and Experimental Methodologies. XII, 530 pages. 2007.

Vol. 4613: F.P. Preparata, Q. Fang (Eds.), Frontiers in Algorithmics. XI, 348 pages. 2007.

Vol. 4600: H. Comon-Lundh, C. Kirchner, H. Kirchner (Eds.), Rewriting, Computation and Proof. XVI, 273 pages. 2007.

Vol. 4599: S. Vassiliadis, M. Bereković, T.D. Hämäläinen (Eds.), Embedded Computer Systems: Architectures, Modeling, and Simulation. XVIII, 466 pages. 2007.

Vol. 4598: G. Lin (Ed.), Computing and Combinatorics. XII, 570 pages. 2007.

Vol. 4596: L. Arge, C. Cachin, T. Jurdziński, A. Tarlecki (Eds.), Automata, Languages and Programming. XVII, 953 pages. 2007.

Vol. 4595: D. Bošnački, S. Edelkamp (Eds.), Model Checking Software. X, 285 pages. 2007.

Vol. 4590: W. Damm, H. Hermanns (Eds.), Computer Aided Verification. XV, 562 pages. 2007.

Vol. 4588: T. Harju, J. Karhumäki, A. Lepistö (Eds.), Developments in Language Theory. XI, 423 pages. 2007.

Vol. 4583: S.R. Della Rocca (Ed.), Typed Lambda Calculi and Applications. X, 397 pages. 2007.

Vol. 4580: B. Ma, K. Zhang (Eds.), Combinatorial Pattern Matching. XII, 366 pages. 2007.

Vol. 4576: D. Leivant, R. de Queiroz (Eds.), Logic, Language, Information and Computation. X, 363 pages. 2007.

Vol. 4547: C. Carlet, B. Sunar (Eds.), Arithmetic of Finite Fields. XI, 355 pages. 2007.

Vol. 4546: J. Kleijn, A. Yakovlev (Eds.), Petri Nets and Other Models of Concurrency – ICATPN 2007. XI, 515 pages. 2007.

Vol. 4545: H. Anai, K. Horimoto, T. Kutsia (Eds.), Algebraic Biology. XIII, 379 pages. 2007.

Vol. 4533: F. Baader (Ed.), Term Rewriting and Applications. XII, 419 pages. 2007.

Vol. 4528: J. Mira, J.R. Álvarez (Eds.), Nature Inspired Problem-Solving Methods in Knowledge Engineering, Part II. XXII, 650 pages. 2007.

Vol. 4527: J. Mira, J.R. Álvarez (Eds.), Bio-inspired Modeling of Cognitive Tasks, Part I. XXII, 630 pages. 2007.

Vol. 4525: C. Demetrescu (Ed.), Experimental Algorithms. XIII, 448 pages. 2007.

Vol. 4514: S.N. Artemov, A. Nerode (Eds.), Logical Foundations of Computer Science. XI, 513 pages. 2007.

Vol. 4513: M. Fischetti, D.P. Williamson (Eds.), Integer Programming and Combinatorial Optimization. IX, 500 pages. 2007.

Vol. 4510: P. Van Hentenryck, L.A. Wolsey (Eds.), Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. X, 391 pages. 2007.

Vol. 4507: F. Sandoval, A.G. Prieto, J. Cabestany, M. Graña (Eds.), Computational and Ambient Intelligence. XXVI, 1167 pages. 2007.

Vol. 4502: T. Altenkirch, C. McBride (Eds.), Types for Proofs and Programs. VIII, 269 pages. 2007.

Vol. 4501: J. Marques-Silva, K.A. Sakallah (Eds.), Theory and Applications of Satisfiability Testing – SAT 2007. XI, 384 pages. 2007.

Vol. 4497: S.B. Cooper, B. Löwe, A. Sorbi (Eds.), Computation and Logic in the Real World. XVIII, 826 pages. 2007.

Vol. 4494: H. Jin, O.F. Rana, Y. Pan, V.K. Prasanna (Eds.), Algorithms and Architectures for Parallel Processing. XIV, 508 pages. 2007.

Vol. 4493: D. Liu, S. Fei, Z. Hou, H. Zhang, C. Sun (Eds.), Advances in Neural Networks – ISNN 2007, Part III. XXVI, 1215 pages. 2007.

Vol. 4492: D. Liu, S. Fei, Z. Hou, H. Zhang, C. Sun (Eds.), Advances in Neural Networks – ISNN 2007, Part II. XXVII, 1321 pages. 2007.

Vol. 4491: D. Liu, S. Fei, Z.-G. Hou, H. Zhang, C. Sun (Eds.), Advances in Neural Networks – ISNN 2007, Part I. LIV, 1365 pages. 2007.

Preface

This volume contains the papers presented at the 14th International Symposium on String Processing and Information Retrieval (SPIRE), held in Santiago, Chile, on October 29–31, 2007. SPIRE 2007 was organized in tandem with the 5th Latin American Web Congress (LA-WEB), with both conferences sharing a common day on Web Retrieval.

The papers in this volume were selected from 77 papers submitted from 25 different countries in response to the Call for Papers. Due to the high quality of the submissions, a total of 27 papers were accepted as full papers, yielding an acceptance rate of about 35%. SPIRE 2007 also featured three talks by invited speakers: Andrew Tomkins (Yahoo! Research, USA), Nivio Ziviani (Federal University of Minas Gerais, Brazil) and Justin Zobel (NICTA, Melbourne, Australia).

The SPIRE annual symposium provides an opportunity for researchers to present original contributions on areas such as *string processing* (dictionary algorithms, text searching, pattern matching, text compression, text mining, natural language processing, and automata based string processing), *information retrieval* (IR modeling, indexing, ranking and filtering, interface design, visualization, cross-lingual IR systems, multimedia IR, digital libraries, collaborative retrieval, and Web related applications), *interaction of biology and computation* (DNA sequencing and applications in molecular biology, evolution and phylogenetics, recognition of genes and regulatory elements, and sequence driven protein structure prediction), and *information retrieval languages and applications* (XML, SGML, information retrieval from semi-structured data, text mining, and generation of structured data from text).

Special thanks are due to the members of the Program Committee and the additional reviewers who worked very hard to ensure the timely review of all submitted manuscripts. Thanks are due to Fabiano Cupertino Botelho, a Ph.D. student volunteer who ran the OpenConf system during the reviewing process and helped with the editorial work for this volume. We also thank the local organizers for their support and organization of SPIRE, in particular Javier Velasco, Christian Middleton, and Sara Quiñones, as well as the local team of student volunteers, whose efforts ensured the smooth organization and running of the event.

We would like to thank the sponsoring institutions, the Millennium Nucleus Center for Web Research of the Dept. of Computer Science of the University of Chile, the Dept. of Computer Science of the Federal University of Minas Gerais and Yahoo! Research Latin America.

October 2007

Nivio Ziviani
Ricardo Baeza-Yates

SPIRE 2007 Organization

General Chair

Ricardo Baeza-Yates	Yahoo! Research (Spain & Chile) and CWR/DCC, Universidad de Chile (Chile)

Program Committee Chair

Nivio Ziviani	Universidade Federal de Minas Gerais (Brazil)

Local Organization

Fabiano C. Botelho	Universidade Federal de Minas Gerais (Brazil)
Christian Middleton	Universitat Pompeu Fabra (Spain)
Sara Quiñones	Yahoo! Research Latin America (Chile)
Javier Velasco	CWR/DCC, Universidad de Chile (Chile)

Steering Committee

Alberto Apostolico	Università di Padova (Italy) and Georgia Tech (USA)
Ricardo Baeza-Yates	Yahoo! Research (Spain & Chile) and CWR/DCC, Universidad de Chile (Chile)
Mariano Consens	University of Toronto (Canada)
Fabio Crestani	Università della Svizzera Italiana (Switzerland)
Paolo Ferragina	Università di Pisa (Italy)
Massimo Melucci	Università di Padova (Italy)
Gonzalo Navarro	Universidad de Chile (Chile)
Berthier Ribeiro-Neto	Universidade Federal de Minas Gerais (Brazil)
Mark Sanderson	University of Sheffield (UK)
Nivio Ziviani	Universidade Federal de Minas Gerais (Brazil)

Program Committee Members

James Allan	University of Massachusetts Amherst (USA)
Amihood Amir	Bar-Ilan University (Israel)
Alberto Apostolico	University of Padova (Italy) and Georgia Tech (USA)
Chris Buckley	Sabir Research (USA)

Pável Pereira Calado	Instituto Superior Técnico/INESC-ID (Portugal)
Maxime Crochemore	University of Marne-la-Vallée (France)
Bruce Croft	University of Massachusetts Amherst (USA)
Martin Farach-Colton	Rutgers University (USA)
Edward Fox	Virginia Tech (USA)
Kimmo Fredriksson	University of Joensuu (Finland)
Raffaele Giancarlo	University of Palermo (Italy)
Marcos André Gonçalves	Federal University of Minas Gerais (Brazil)
Roberto Grossi	University of Pisa (Italy)
Heikki Hyyrö	University of Tampere (Finland)
Lucian Ilie	University of Western Ontario (Canada)
Costas Iliopoulos	University of London (UK)
Juha Kärkkäinen	University of Helsinki (Finland)
Mounia Lalmas	University of London (UK)
Tak-Wah Lam	University of Hong Kong (Hong Kong)
Gad Landau	University of Haifa (Israel)
Thierry Lecroq	University of Rouen (France)
Andrew MacFarlane	City University London (UK)
Veli Mäkinen	University of Helsinki (Finland)
Giovanni Manzini	University of Piemonte Orientale (Italy)
Massimo Melucci	University of Padua (Italy)
Alistair Moffat	University of Melbourne (Australia)
Edleno Silva de Moura	Federal University of Amazonas (Brazil)
Ian Munro	University of Waterloo (Canada)
Gonzalo Navarro	University of Chile (Chile)
Arlindo Oliveira	Inst. Superior Técnico/INESC-ID/IST (Portugal)
Sándor Pongor	Intl. Centre for Genetic Eng. and Biotechnology (Italy)
Bruno Pôssas	Google Inc.(Brazil)
Mathieu Raffinot	CNRS (France)
Kunihiko Sadakane	Kyushu University (Japan)
Marie-France Sagot	INRIA and University Claude Bernard, Lyon I (France)
João Setubal	Virginia Tech (USA)
Rahul Shah	Purdue University (USA)
Altigran Soares da Silva	Federal University of Amazonas (Brazil)
Fabrizio Silvestri	ISTI - CNR (Italy)
Wing-Kin Sung	National University of Singapore (Singapore)
Masayuki Takeda	Kyushu University (Japan)
Jorma Tarhio	Helsinki University of Technology (Finland)
Gabriel Valiente	Technical University of Catalonia (Spain)
Hugo Zaragoza	Yahoo! Research (Spain)
Justin Zobel	RMIT University (Australia)

Additional Reviewers

José Ramón Pérez Agüera
Guilherme Assis
Marie-Pierre Béal
Ido Dagan
Chiara Epifanio
Iman Hajirasouliha
Wing-Kai Hon
Tomi Klein
Jorma Laurikkala
Turkka Näppilä
Simon Puglisi
Leena Salmela
Tuomas Talvensaari
Jarkko Toivonen
Sebastiano Vigna
Diego Arroyuelo
Claudine Badue
Mathieu Constant
Gianna Del Corso
Ankur Gupta
Jan Holub
Petri Kalsi
Alberto H. F. Laender
Sabrina Mantaci
Hannu Peltola
Cenk Sahinalp
Borkur Sigurbjornsson
Andrew Turpin
Shuly Wintner

Previous Venues of SPIRE

The first four editions focused primarily on *string processing* and were held in Brazil and Chile. At that time SPIRE was called WSP (South American Workshop on String Processing). Starting in 1998, the focus of the workshop was broadened to include the area of *information retrieval*, due to the latter's increasing relevance and its inter-relationship with the area of string processing, and the name of the workshop was changed to the current one. In addition, since 2000, the symposium has been held alternately in Europe and Latin America, and has so far been held in Mexico, Spain, Chile, Portugal, Brazil, Italy, Argentina and the UK.

2006: Glasgow, UK
2005: Buenos Aires, Argentina
2004: Padova, Italy
2003: Manaus, Brazil
2002: Lisboa, Portugal
2001: Laguna San Rafael, Chile
2000: A Coruña, Spain
1999: Cancun, Mexico
1998: Santa Cruz de la Sierra, Bolivia
1997: Valparaiso, Chile
1996: Recife, Brazil
1995: Viña del Mar, Chile
1993: Belo Horizonte, Brazil

Table of Contents

A Chaining Algorithm for Mapping cDNA Sequences to Multiple Genomic Sequences

Mohamed Abouelhoda

Faculty of Engineering and Computer Science, Ulm University
D-89069 Ulm, Germany
mohamed.ibrahim@uni-ulm.de

Abstract. Given a set of matches between a cDNA sequence and *multiple* genomic sequences, we present a subquadratic chaining algorithm for computing an optimal chain of colinear matches, while allowing overlaps between the matches. Our algorithm improves upon the quadratic graph based solution, and extends the previous algorithms which are limited to matches between a cDNA sequence and a *single* genomic sequence. The matches of the resulting chain serve as anchors for computing a multiple alignment between the cDNA and the given sequences.

1 Introduction

A fundamental task of every genome annotation project is to locate each gene in the genome and to determine its structure. This knowledge serves as a basis for elucidating the gene function and studying the genome organization and evolution. One of the most successful methods for accomplishing this task is the mapping of cDNA sequences to the genomes they are transcribed from. A cDNA sequence is a complementary sequence to a mRNA. Because the introns are spliced out from a mRNA and just the exons remain, an alignment of a cDNA to the related genomic sequence locates the corresponding gene and directly reveals its exon-intron structure; see Figure 1 (a). The increasing number of full cDNA sequencing projects reflects the growing popularity of this method.

For high throughput mapping of cDNA sequences, standard dynamic programming algorithms are impractical due to their quadratic running time. Hence, heuristic algorithms have been developed; see e.g. [7,8,12] and the references therein. Most of these tools use an *anchor-based strategy* composed of three phases: (1) computation of fragments (regions in the sequences that are similar), (2) computation of an optimal chain of colinear fragments; these are the anchors that form the basis of the alignment, (3) alignment of the regions between the anchors considering the splice site signals.

The algorithm of Shibuya and Kurochkin [12] is superior to other ones because of two novel improvements: First, the fragments are of the type (rare) maximal exact match computed by means of the suffix tree of the genomic sequence in linear time and space. Second, in contrast to other algorithms, their chaining algorithm is geometry based and allows overlaps between the fragments.

N. Ziviani and R. Baeza-Yates (Eds.): SPIRE 2007, LNCS 4726, pp. 1–13, 2007.

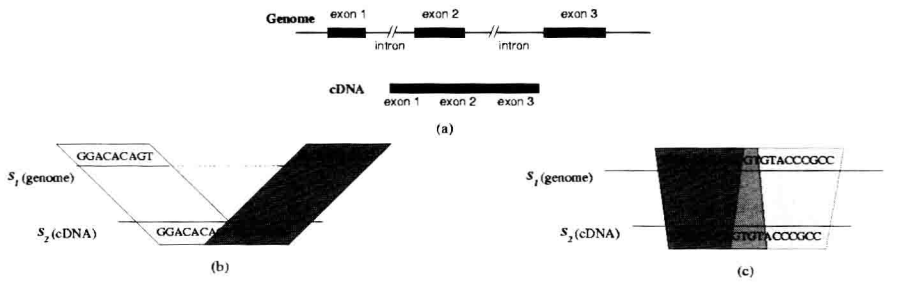

Fig. 1. (a): A cDNA mapped to a genomic sequence. The exons are separated by long introns in the genome. (b): Fragments (represented by parallelograms) overlap in the cDNA sequence only. (c) The overlap is in both the cDNA and the genome.

(The overlap lengths are taken into account, i.e., penalized, in the objective function.) Their chaining algorithm takes $O(m \log m)$ time and requires $O(m)$ space, where m is the number of the fragments. (Algorithms permitting *no* overlaps have been presented in [1,6,10,13].) Although this chaining algorithm is relatively complicated due to the combination of range maximum queries and the candidate list paradigm, it is an important improvement over the naive graph based solution that takes $O(m^2)$ time [12].

The rationale behind permitting overlaps is twofold: First, overlapping fragments were found to be very common in cDNA mapping [7,12], and they usually occur at the exon boundaries in the cDNA; see Figure 1 (b). Second, the amount of sequence covered by the chain will increase, which is crucial for both improving the sensitivity/specificity and for speeding-up the mapping task. Regarding the sensitivity/specificity, some fragments may be discarded as a result of permitting no overlap in the chain. This can reduce the chain coverage under the threshold defined by the user to filter out noisy chains, and consequently results in discarding the whole chain despite its potential significance. If one attempts to overcome this drawback by decreasing the threshold, many insignificant chains will not be filtered out and the specificity will decrease. Regarding the running time, the less the chain coverage, the higher the running time of the third phase in which an alignment on the character level is computed to finish the mapping task.

The genomes of very closely related species or the genomes of different strains of the same species share a very large sequence identity, and so does a cDNA sequence to the genome it stems from. Therefore, it is natural to extend the algorithm of Shibuya and Kurochkin to map a cDNA sequence to multiple genomes. Such an extension, in addition to the theoretical interest related to it, will help in both identifying the common genes among the genomes and determining the syntenic regions (regions of conserved gene-order) among the genomic sequences.

This extension, however, is not straightforward. While computing fragments from multiple genomic sequences can be easily achieved in linear time and space [2,5,11,9], the extension of the chaining algorithm of Shibuya and Kurochkin to chain fragments from k sequences while permitting overlaps is extremely

complicated, if not infeasible. This is due to the difficulty of analyzing the overlaps, according to the objective function they suggested, and due to the difficulty of combining the range queries and candidate lists; Shibuya and Kurochkin noticed also these complications [12].

In this paper we handle the combinatorial chaining problem with overlap for mapping a cDNA sequence to multiple genomic sequences. We show in this paper that an efficient subquadratic chaining algorithm exists, if an objective function specific to the cDNA mapping task is used. We present this algorithm, and, moreover, address two special cases of practical interest: (1) the usage of rare *multi-MEMs*, and (2) constraining the amount of overlap. For these two cases, we show that the algorithm complexity can be further improved. Our algorithms are easy to implement, because they use solely range queries without any candidate lists. They are also so efficient that millions of fragments are processed in a few minutes.

In the following section, the definitions are stated. Section 3 introduces the chaining problem and the graph based solution. In Section 4, we present our geometry based algorithm. In Section 5, we focus on two special cases of the basic algorithm. Sections 6 and 7 contain experimental results and conclusions.

2 The Fragments

2.1 Definitions

For $1 \leq i \leq k$, S_i denotes a string of length $|S_i|$. In our application, S_i represents a cDNA or a genomic sequence. $S_i[h_1..h_2]$ is the substring of S_i starting at position h_1 and ending at position h_2, and $S_i[h_1]$ denotes the h_1^{th} character of S_i, $1 \leq h_1 \leq h_2 \leq |S_i|$. A *fragment* is a region of similarity among the given sequences. In this paper, we use fragments of the type *(rare) maximal multiple exact match*, denoted by *(rare) multi-MEM* and defined as follows.

A *multiple exact match* among k sequences $S_1, \ldots, S_k$ is a $(k+1)$-tuple $(l, p_1, .., p_k)$ such that $S_1[p_1..p_1+l-1] = \ldots = S_k[p_k..p_k+l-1]$; i.e., the l-character-long substrings of $S_1, \ldots, S_k$ starting at positions $p_1, \ldots, p_k$, respectively, are identical. A multiple exact match is *left maximal* if $S_i[p_i - 1] \neq S_j[p_j - 1]$ for any $1 \leq i \neq j \leq k$, and *right maximal* if $S_i[p_i + l] \neq S_j[p_j + l]$ for any $1 \leq i \neq j \leq k$, i.e., it cannot be extended to the left and to the right simultaneously in all the sequences. A *multi-MEM* is a left and right maximal multiple exact match.

A *multi-MEM* $(l, p_1, .., p_k)$ is called *rare*, if the substring $S_i[p_i..p_i+l-1]$ occurs at most r times in each S_i, $1 \leq i \leq k$. A *maximal multiple unique match* (*multi-MUM*) is a rare *multi-MEM* such that $r = 1$, i.e., $S_i[p_i..p_i+l-1]$ occurs exactly once in each S_i.

A hyper-rectangle in a k dimensional space ($\mathbb{R}^k$) can be represented by the k-tuple $([p_1..q_1], \ldots, [p_k..q_k])$, where $[p_i..q_i]$ is the interval on the coordinate axis x_i, $1 \leq i \leq k$. Equivalently, this hyper-rectangle can be denoted by $R(p, q)$, where $p = (p_1, .., p_k)$ and $q = (q_1, .., q_k)$ are its two extreme corner points. A fragment of the type (rare) *multi-MEM* $(l, p_1, .., p_k)$ can be represented by a hyper-rectangle in $\mathbb{R}^k$ with the two extreme corner points $(p_1, .., p_k)$ and

$(p_1 + l - 1, .., p_k + l - 1)$. In the following, we will denote these corner points by $beg(f) = (beg(f).x_1, .., beg(f).x_k)$ and $end(f) = (end(f).x_1, .., end(f).x_k)$, respectively. Furthermore, we define $f.length = l$ to denote the length of the *multi-MEM* corresponding to f.

Throughout this paper, the k^{th} sequence is the cDNA sequence. For ease of presentation, we consider the point $0 = (0, \ldots, 0)$ (the origin) and the terminus $t = (|S_1| - 1, \ldots, |S_k| - 1)$ as fragments with length zero.

2.2 Computing the Fragments

Computing (rare) *multi-MEMs* between $k - 1$ genomic sequences and a cDNA database can be achieved in a linear time and space. One strategy is to proceed as follows: Construct the sequence S_k by appending unique characters to each cDNA and concatenating all of them. Construct the sequence $\hat{S}$ by appending unique characters to each genomic sequence and S_k and concatenating all of them. Then build the suffix tree (or the enhanced suffix array [2]) for $\hat{S}$. A *multi-MEM* $(l, p_1, .., p_k)$ is a match in $\hat{S}$ such that, $p_1 \in [1..(|S_1| + 1])$, $p_2 \in [(|S_1| + 2)..(|S_1| + |S_2| + 2)], \ldots$ and $p_k \in [(|S_1| + \ldots + |S_{k-1}| + k)..(|S_1| + \ldots + |S_k| + k)]$. Computing *multi-MEMs* can be achieved by a bottom-up traversal of the suffix tree of S_k, as described in [2]. There it is also shown that the rareness constraint can be satisfied during the traversal without extra cost (the rareness value w.r.t. S_i in [2] is the value $C_{\mathcal{P}}(S_i)$). For *multi-MUMs*, the algorithm in [5] requires a single scan of the enhanced suffix array, and it is easy to implement.

A more efficient strategy for computing (rare) *multi-MEMs* has recently been developed [11]. The idea is to construct the suffix tree (or the enhanced suffix array) for the shortest genomic sequence, say S_1. Then the remaining genomic sequences $S_2, \ldots S_{k-1}$ are sequentially matched against the suffix tree using the Chang-Lawler Algorithm [4]. During this matching, nodes of the suffix tree are annotated with match information. Only the nodes satisfying the rareness constraint are taken into account. Then the cDNA database is queried against the annotated suffix tree to further annotate more nodes. Finally, all (rare) *multi-MEMs* are reported through a bottom-up traversal of the suffix tree. The program `ramaco` is an implementation of the algorithm in [11]. The program M-GCAT [9], although no details are given, seems to use a similar approach for computing *multi-MUMs*.

3 Chaining Fragments with Overlaps

Definition 1. *Let f' and f be two fragments with $beg(f').x_i < beg(f).x_i$, for all $1 \leq i \leq k$. We say that f'* overlaps *with f in S_i iff (1) $end(f').x_i < end(f).x_i$ for all $1 \leq i \leq k$, and (2) $end(f').x_i \geq beg(f).x_i$, for any $1 \leq i \leq, k$.*

For $k = 2$, Figure 1 (b) shows two fragments overlapping in S_2 but not in S_1, while Figure 1 (c) shows two fragments overlapping in both S_1 and S_2.

Definition 2. *The relation $\ll$ on the set of fragments is defined as follows. $f' \ll f$ iff the following two conditions hold: $beg(f').x_i < beg(f).x_i$ and $end(f').x_i <$*

$end(f).x_i$, for all $1 \leq i \leq k$. If $f' \ll f$, then we say that f' precedes *f. The fragments f' and f are* colinear *if either f' precedes f or f precedes f'.*

Thus, two fragments are colinear if they appear in the same order in all sequences. Note that if we further have $end(f').x_i < beg(f).x_i$, for all $1 \leq i \leq k$, then f' and f are colinear and non-overlapping. A geometric representation of this relation for $k = 2$ is given in Figure 2 (a), where any fragment $f' \ll f$ must start in Region $A(f)$ and end in region $\{AB(f) \cup C(f)\}$; $A(f)$ is the rectangular region $R(0, beg(f))$, $AB(f)$ is the rectangle $([0..end(f).x_1 - 1], [0..beg(f).x_2 - 1])$, and $C(f)$ is the region $([0..end(f).x_1 - 1], [beg(f).x_2..end(f).x_2 - 1])$. For $k > 2$, $AB(f)$ and $C(f)$ are the hyper-rectangles $([0..end(f).x_1 - 1], .., [0..end(f).x_{k-1} - 1], [0..beg(f).x_k - 1])$, and $([0..end(f).x_1 - 1], .., [0..end(f).x_{k-1} - 1], [beg(f).x_k ..end(f).x_k - 1])$, respectively.

Definition 3. *For any two fragments f and f' from k sequences, where the k^{th} sequence is the cDNA sequence, the amount of overlap in the cDNA sequence is*

$$overlap_k(f', f) = \begin{cases} end(f').x_k - beg(f).x_k + 1, & \text{if } beg(f).x_k \leq end(f').x_k \leq end(f).x_k \\ 0, & \text{otherwise} \end{cases}$$

Accordingly, the cDNA chaining problem can be formulated as follows.

Definition 4. *Given a set of m fragments, find a chain C of colinear fragments $f_1, f_2, .., f_t$ (i.e., $f_1 \ll f_2 \ll .. \ll f_t$) such that $score(C) = \sum_{i=1}^{t} f_i.length - \sum_{i=1}^{t-1} overlap_k(f_i, f_{i+1})$ is maximal.*

This objective function penalizes the overlaps and maximizes the amount of cDNA sequence mapped to the genomic sequence; which is the target of the cDNA mapping problem. It is easy to see that a perfect mapping has a score that equals the cDNA length. As we will show later in our geometry based solution, this objective function has the advantage that for each fragment f only two regions ($AB(f)$ and $C(f)$) are considered, independently of k, when constructing an optimal chain.

A straightforward solution to the cDNA chaining problem is to construct a weighted directed acyclic graph $G(V, E)$, where the set of vertices V is the set of fragments (including 0 and t), and the set of edges E is characterized as follows. For any two nodes $v' = f'$ and $v = f$, there is an edge $e(v' \rightarrow v) \in E$ with weight of $f.length - overlap(f', f)$, only if $f' \ll f$; see Figure 2 (b). An optimal chain corresponds to a path with maximum score from vertex 0 to vertex t in the graph. Because the graph is acyclic, such a path can be computed as follows. Let $f.score$ denote the maximum score of all chains ending with the fragment f. Clearly, $f.score$ can be computed by the recurrence

$$f.score = f.length + \max\{f'.score - overlap_k(f', f) | f' \ll f\} \quad (1)$$

A dynamic programming algorithm based on this recurrence takes $O(m^2)$ time, where m is the number of fragments. However, this quadratic running time is a drawback for a large number of fragments. In the following section, we present a geometry based solution that runs in subquadratic time.